### Ajax Nedir?

AJAX (Asynchronous Javascript and XML) birileri tarafından geliştirilmiş bir teknoloji olmayıp, daha çok bir yöntemdir ve içerisinde bir çok teknolojiyi barındırır. Bir yöntem olarak ajax, internet tarayıcılarının asenkron olarak sunucudan istekte bulunmasını sağlayarak web uygulamalarının kullanıcı arabirimini masaüstü uygulamalara bir adım daha yaklaştırır. Geleneksel web uygulamalarında, internet tarayıcıları sunucudan, sayfanın tüm içeriğini getirmesi için istekte bulunur. Kullanıcı gelen sayfa üzerinde bulunan bir bağlantıyı tıklayarak ya da doldurduğu bir formu göndererek yeni bir istekte bulunabilir. Sunucu gelen bu yeni istek için yine kullanıcıya sayfanın tamamını gönderir. Bu durum Şekil 1 de gösterilmiştir



. Şekil.1 Geleneksel yapıda web sayfasın isteği

Ajax tekniğinde ise, Şekil 1’de gösterilen durum farklıdır. Kullanıcının tıkladığı bağlantı veya gönderdiği forma karşılık, sayfanın tamamı yerine mevcut sayfa içerisinde değişmesi gereken içeriğin bulunduğu bölüm, gelen cevap ile güncellenir. Bu durum Şekil 2’de gösterilmiştir. Şekil 2’de gösterilen yapıyı gerçekleştirebilme, başka bir değişle sayfanın ilgili yerlerinin güncellenebilmesini sağlayabilmek için Microsoft’un geliştirdiği, XMLHttpRequest nesnesini kullanan bir arabirime ihtiyaç vardır.



Şekil.2 AJAX Tekniği İle Web Sayfasının İsteği

Şekil 3,4’de geleneksel yolun ajax tekniği ile karşılaştırılması zaman ekseninde gösterilmiştir. Şekil 3,4 den açıkça anlaşılacağı gibi geleneksel web modelinde her bir kullanıcı aktivitesinde senkron olarak sunucudan istemciye veri akışı mevcut iken ajax web uygulama modelinde veri akışı asenkrondur. Bu tür yaklaşım ile gereksiz veri trafiğinin önüne geçilmiştir. Yukarıda belirtilen “senkron” ve “asenkron” ifadeleri zaman ekseninde senkronizasyonu belirtmemektedir. Bu çalışmada “senkron” ile istemcide gösterilen sayfanın bütünü ile sunucudan alındığı ve bu sürede tarayıcının başka bir iş yapmadan bu isteğin cevabını beklediği, “asenkron” ile istemcide gösterilen sayfanın, herhangi bir zaman diliminde bir kısmının sunucudan alınması ve tarayıcının bu işlemler sırasında diğer işlemleri yapabilmesine olanak sağlaması kast edilmiştir.

AJAX tekniğinin kullandığı teknolojiler, aşağıda sıralanmıştır.

• HTML/XHTML: Sayfanın içeriğinin görüntülenmesini sağlar.

• DOM: Sayfa içerisindeki kontrollere ulaşabilmesini ve içeriklerinin değiştirilebilmesini sağlar.

• XMLHttpRequest: Sunucu ile tarayıcının iletişimi sağlayan nesnedir. Bu nesne sayfanın tümünün istemciye sunucu tarafından gönderilmeden haberleşilebilmesini sağlar.

• JavaScript: AJAX Teknolojisini uygulamak için kullanılan script dilidir.

• XML,JSON: Sunucu ve istemci arasındaki veri iletişimini sağlayan modalardır.

• XSLT: XML tabanlı stil dosyasıdır.

• CSS: Sayfadaki öğelere biçim ve sitiller uygulamamızı sağlar.

Ajax Mimarisi

Klasik bir istemci-sunucu ( client-server) uygulamasında kullanıcı tarafından yapılan her işlem sunucuya HTTP Request olarak gönderilir , yorumlanır ve veriler işletildikten sonra sonuçlar istemciye gönderilir. Bu da çok büyük bir ağ trafiğine neden olur. AJAX kullanıldığında ise istekler HTTP REQUEST yerine XMLHTTP REQUEST olarak gönderilir. Burada sunucu ve istemci arasında taşınan veriler sıkıştırılmış XML formatındadır. Bu sıkıştırılmış yapı istemci tarafında açılır böylece sunucu ve istemci arasındaki bant genişliği boş yere işgal edilmemiş olur. AJAX ile sağlanan en büyük özellik de PARÇALI GÜNCELLEME’ dir. Bu sayade bütün bir sayfanın istemci ile sunucu arasında gidip gelmesiyle doğan ağ trafiğinin önüne geçilmiş olur .Önceden Meta taglerine refresh yazarak bu işlemi yaparılırdı fakat bütün sayfa güncellenirdi ve kullanıcı boş bir sayfaya ile başbaşa kalırdı.

Ajax Tarihçesi

AJAX ile Amerika yeniden keşfedilmemiştir. Yıllardır kullanılan JAVASCRIPT ve XMLbirleştirilerek yeni bir uygulama deseni oluşturulmuştur. Bu düşünce önce JSP geliştiricileri tarafından ortaya atılmış daha sonra da Microsoft tarafından desteklenmiştir. Önceleri Microsoft bunu ATLAS olarak adlandırmış daha sonra da ASP.NET AJAX olarak desteğini sürdürmüştür. Burada Javascript denilince farklı tarayıcılarda farklı sonuçların ortaya çıkması gibi bir durum söz konusudur. Maalesef internet tarayıcılarımızı üreten firmalar W3C standartalarına sağdık kalmamaktalar ve bu da Javascriptlerin farklı sonuçlar üretmesine neden olmaktadır. Microsoft ASP.NET AJAX mimarisinin içine istemciler için “Browser Compatibility” kütüphanesini de dahil etmiştir. AJAX ile ilgili olarak arama motorlarının siteyi indekslemesi sırasında da sorunlara neden olduğu haberleri ortalıkta dolaşmaktadır. Artık Web 2.0 ile siteler kullanıcılarına daha iyi bir web deneyimi yaşatmayı amaçlıyorlar ve AJAX da bunun için iyi bir araç

Ajax Nerede Kullanılır

Ajax hemen hemen web’de bir çok şey için kullanılır. Bunun bir örneği ise Google Maps. Kullanıcı sayfadakı haritaya tıklayıp resmi çektiği an, bilgi anında gösterilir butun sayfa yeniden yüklenmeyerek. Sadece bilgi yüklenir görüntü kırpmadan. Bütün bunlar Ajax’ın senkronos özelliği tarafından yapılmaktadır. Ajax aynı zamanda Gmail’de bile kullanılmaktadır. Bilginin yüklenmesi gerekmiyor, çünkü uygulamalar Ajax kullanarak etkileşmektedir ki sayfanın bütününün yeniden yüklenilmemesi için. Yeni posta geldiğinde, hemen otomatik gelen kutuya eklenir. Bu kullanıcının yenileme düğmesinin gerekli olmadığını sağlar. Google Translate bile Ajax kullanmaktadır. Kullanıcılar bır yazıyı bir dilden başka bir dile tercüme etmek için yazmağa başladıkları zaman, Ajax hemen diğer dildeki analamı aynı sayfada gösterir. Google dışında ajax kullanan Flickr ve Amazon da vardır.

Ajax’ın Artıları ve Eksileri

Artılar:

+ AJAX' web sayfalarına kullanım kolaylığı ve etkileşim özelliği (responsiveness) getirmiştir.

+ AJAX ile birlikte sayfadaki verinin tamamının sunucuya gidip gelmesine gerek olmaz, sadece küçük parçalar gidip gelir. Bu anlamda band genişliğinin daha efektif kullanımını sağlar.

+ AJAX programlama dillerinden bağımsız bir teknolojidir; Java, PHP yada .NET ortamlarında aynı şekilde kullanılabilir.

+ Asenkron çalışma: Yani AJAX sayesinde client, sunucudan gelen veri akışının bitmesini beklemeksizin işlemlerine devam edebilir.

Eksiler:

- Browser uyumsuzluğu. AJAX ile yazdığınız kod IE'de çok güzel çalışırken Firefox'ta tamamen işlevsiz olabiliyor ya da tam tersi Ama geliştirilen AJAX kütüphanelerini kullanarak (prototype gibi) bu sorunu belirli bir düzeye indirebilirsiniz

- Geri tuşu ya da sayfayı imlere ekleme gibi tarayıcı özelliklerini kullanamamamız

- URL kopyala, yapıştırma :Genel kullanıcı alışkanlığı site içeriğini URL olarak saklamak istiyorsa ve site AJAX ile geliştirilmişse bunu yapması mümkün olamaz.

- Arama motorları web sayfasındaki içeriğe göre indeksleme yapar. Ajax ile sunuduğumuz içerik ise arama motorları tarafından görüntülenemez.

- AJAX ile geliştirilmiş içerikleri tarayıcıların yazdırması sorun olabiliyor.

- AJAX uyumlu olmayan browserları kullananlar, yada Javascripti engellemiş kullanıcılar AJAX ile sunduğunuz içeriğe ulaşamazlar.

- Sayfanın tepkisiz görünebilmesi. AJAX ile gelen-giden veri miktarı büyükse yada client-server arası bağlantı hızı yavaş ise, kullanıcı işlem yapmasına rağmen browser'dan beklediği tepkiyi alamaz, sayfada hiçbir değişiklik olmadığı için sitenin kitlendiği izlenimine kapılılabilir. Bunu mutlaka uyarı mesajlarıyla engellemek gerekir.

HTML DOKÜMAN OBJE MODELİ (HTML DOM):

Ajax tekniğinde dinamik içerik güncelleme ve kullanıcı etkileşimi için DOM(Document Object Model) kullanıldığından DOM ile alakalı bir bölümün hazırlanma ihtiyacı, bu tez çalışmasında geliştirilen çoktan seçmeli sınav sisteminin anlaşılabilmesi için oluşmuştur. DOM HTML dokümanlarına erişilebilmesi ve bunlar üzerinde değişiklik yapabilmesi için geliştirilen W3(Word Wide Web Konsorsiyumu) standartlarına uygun bir standartlar topluluğudur. Bir başla değişle, DOM, arayüzün bir parçası olarak, programların ve scripterin dinamik olarak doküman üzerindeki stil, yapı ve içeriklerine erişilebilmesini ve bunların güncellenebilmesini sağlar. Böylelikle DOM standardına göre geliştirilmiş doküman akıllı bir belge haline dönüşür ve ilgili doküman hem daha fazla işlem yapabilir hemde işlem sonuçlarını sunulmuş olan sayfa içerisinde gösterebilir

DOM temel olarak üç seviyeye ayırabilir. Bunlar:

• Core DOM : Yapısal dokümanlar için standart model

• XML DOM : XML dokümanları için standart model

• HTML DOM : HTML dokümanları için standart model

Şeklindedir. Aşağıda her bir seviye ayrı ayrı açıklanmıştır.

3.1. Core DOM

Core DOM, DOM için ana fonksiyonları barındırır ve diğer seviyelere barındırdığı bu fonksiyonlara servis eder. Başka bir değişle DOM’ un temel seviyesidir.

3.2. XML DOM

XML DOM, XML dokümanların obje ve özelliklerini tanımlar ve metotların (arabirimlerin) dokümana erişimini sağlar.

3.3. HTML DOM

HTML DOM, HTML için standart obje modeli olup, standart bir programlama dili olarak kullanılabilir. Bu dil platformdan bağımsızdır ve W3 standartlarına uygundur. HTML DOM’ un kullanımına örnek HTML dokümanı Şekil 5.a da verilmiştir HTML DOM kullanımına örnek olarak:

<div id=”ornek”>

a) </div>ümanı

b) var element\_eris= document.layers[“ornek”];//Netscape için

/var element\_eris= document.all[“ornek”];//Internet Explorer için

c) İf (document.all)

{

//internet explorer için kodlar

}

else if(document.layers)

{

//netscape için kodlar

}

(a) HTML örnek kodu

(b) Javascript kullanarak bu koddaki “div” elementine erişebilmek için kullanılan yapı

(c) İki tarayıcıda da çalışabilmesi için şu şekilde kullanılan yapı.

3.4. DOM Metot ve Özellikleri

**DOM metot ve özellikleri iki ana guruba ayrılabilir**. Bunlar sırası ile

----DOM düğümlerinde gezinebilmeyi ve düğüm değer ve özelliklerini çağırabilmeyi sağlayan metot ve özellikler

---- DOM düğümleri ekleyebilmeyi, silebilmeyi, üzerlerinde işlem yapabilmeyi sağlayan metot ve özellikler ‘dir .

Yukarıda belirtilen her iki gurup için belirtilen metot ve özelliklerin DOM destekli herhangi bir dil içerisinde kullanılabilmesi için DOM ağacı için yapacağı işlemlerde aynı metot ve özellikler kullanılmalıdır Aşağıda bazı DOM metot ve özellikleri, bu tez çalışmasında yapılan projenin anlaşılabilmesi için sıralanmıştır.

**getElementByID**

“ornek” id sine sahip “div” elementine DOM ile erişmek için “document” objesinin metotlarından biri olan getElementById metodunu kullanılabilir. Şekil 6’da örnek kullanım gösterilmiştir

var element\_eris = document.getElementById(“ornek”);

**getElementByTagName**

Bu komut getElementById’ye benzer, aralarındaki fark, elementin id si yerine “tag’ının ismi kullanılarak erişim sağlanmasıdır. Bu metodun kullanımı için

Şekil 7’de örnek bir html kodu verilmiştir[1].

<p>merhaba

</p>

<p>dünya

</p>

Şekil 8’de verilen koddaki “p” elementine javascript kullanılarak ulaşmak için; var paragraf=document.getElementByTagName(“p”);

Elemente Erişmek için örnek kod

Şekil 8’ de verilen ifade kullanılabilir.

Paragraf değişkeni, doküman içerisindeki bütün “p” elementlerini içerisinde barındıran bir element dizisidir. Örneğin bu dokümanın içerisinde kaç tane paragraf olduğunu öğrenilmek istenirse

alert(paragraf.length);

Şekil.9 Paragraf sayısının ekrana yazdırılması örneği

9 ifadesinin kullanılması yeterlidir. Eğer dokümanın içerisindeki her bir paragrafa tek tek erişmek istenirse

Şekil 10’ daki kod kullanılabilir.

for(i=0;i<paragraf.length;i++){

paragraf[i] .........

}

**getAttribute**

<p id=”giris” title=”giris mesaji”>

</p>

*getAttribute ile erişilecek örnek html kodu*

getAttribute ile “p” elementinin özniteliklerine erişmek için javascript tarafında

kodların yazılması gerekir.

var elementimiz= document.getElementById(“giris”);

var oznitelik=elementimiz.getAttribute(“title”);

*“title” özelliğine erişen örnek kod*

Bu şekilde “p” elementinin title özniteliği alınmış oluyor.

**Düğümler**

DOM’a göre bir HTML dokümanındaki her şey aslında bir düğüme bağlıdır. Buna Göre

􀀹 Bütün doküman,”document” düğümüdür.

􀀹 Her HTML tagı bir “element” düğümüdür.

􀀹 HTML elementlerinin her bir metni, “text” düğümüdür.

􀀹 Her HTML özniteliği bir “attribute” düğümüdür.

getElementById, getElementByTagName, getElementByAttribute metotları ile aslında

dokümanın düğümlerine erişilmiş olur. Şekil 13’de örnek html dokümanı ve Şekil 14’de de

aynı dokümanın düğüm yapısı verilmiştir.

<html>

<head>

<title>Sayfanin Basligi</title>

</head>

<body>

<h1>DOM dugumu ornegi</h1>

<p>Merhaba Dunya</p>

</body>

</html>

****

Şekil.13 Örnek HTML kodu Şekil.14 Örnek HTML kodunun düğüm yapısı

Ajax örnegi

Ajax 'ın nasıl çalıştığını anlamak için küçük bir ajax uygulaması oluşturacağız.  
İlk önce, username ve time adında iki metin girişi bulunan standart  
bir HTML formu oluşturacağız. username alanı kullanıcı tarafından  
doldurulacak ve time alanı AJAX tarafından doldurulacak.  
Oluşturduğumuz HTML dosyasının adı "testAjax.htm" olsun (bu formda gönder tuşunun olmadığına dikkat edin!):

<html><body>  
<form name="myForm">Name: <input type="text" name="username" />Time: <input type="text" name="time" /></form>  
</body></html>

# Ajax Kütüphaneleri

* Ajax Gear Toolkit  
  Ajax olarak bilinen istemci taraflı tekniğin bütün avantajlarından faydalanmayı mümkün kılan  bir araçtır. XMLHttpRequest nesnelerini kullanarak internet tarayıcısının asenkron çağrı yapmasını sağlar; yani sayfadaki bütün bilgiler sunucuya gitmez. Ajax uygulamalarında olduğu gibi sadece istek yapılan bölüm sunucuya gider.
* Asp.Net Ajax   
  ASP.NET AJAX, ASP.NET ile yapılan ve tüm İnternet tarayıcılarında çalışabilen uygulamalar için Microsoft tarafından geliştirilen bir kütüphanedir. Önce ‘Atlas’, daha sonra ‘ASP.NET  Ajax’ adını almıştır. ASP.NET  2.0 ile kolay entegrasyon, ASP.NET 3.5 ile tam entegrasyon sağlar.
* Behaviour  
  Behaviour,  Xhtml yapılı bir kütüphanedir.
* Dojo Toolkit

Açık kaynak kodlu bir javascript toolkit olup dhtml sorunlarını büyük ölçüde çözmüştür.Arayüz programlama,karmaşık kodların basite indirgenmesi toolkitin sağladığı avantajlardır. Ayrıca dinamik web sitesi ya da başka ortamlar hazırlamayı kolaylaştıran ve sitenin daha kullanışlı olmasını sağlayan içerikler sunar.

* MochiKit

Dreamweaver web uygulamaları için kullanılan ajax kütüphanelerinden birisidir. Eş zamanlı olmayan davranışlar için erteleme uygulamasını kullandığından, sunucuyla etkileşim halinde olan web sayfalarının geliştirilmesinde kullanışlıolduğundan tercih edilir.

* PHPLiveX

PHP ile yapılan uygulamalar için kullanılan bir kütüphanedir. Diğer php tabanlı kütüphaneler gibi php fonksiyonlarını javascript koduyla çağırır; fakat diğerlerine göre daha az kodla kullanımı sağlar. Kullanılan her php fonksiyonu için bir javascript fonksiyonu oluşturur ve ona ek bir argüman girilir. Buraya gerekli değerler yazılıp fonksiyon çağırıldığında javascript ile xmlhttp nesnesi kullanılarak php fonksiyonu çalıştırılır ve dönen değer herhangi bir HTML DOM elemanının içine yazdırılabilir.

* Prototype

Açık kaynaklı ve nesne tabanlı kod kütüphanesidir. Ajax için yazılan kütüphanelerin ilk örneklerindendir ve diğer hazır kütüphanelerin bir kısmı prototype.js kullanmaktadır.

* Sardalya(Sarmal Dinamik Altyapısı)

PHP destekli uygulamalarda kullanılan bir kütüphanedir.Tüm tarayıcılardan bağımsız olarak çalışır ve kullanımı sırasında az yer kaplar.

* Sajax

Açılımı Simple Ajax Toolkit olan Sajax,  ASP ve PHP tabanlı programlarda Ajax uygulamaları geliştirmek için kullanılan bir kütüphanedir. Bütün sunucu taraflı platformlara çağrı gönderebilme yeteneğine sahiptir.

* Solvent

JavaScript ile yazılmıştır ve tüm tarayıcılardan bağımsız olarak çalışan bir Ajax uygulamasıdır. Web uygulamalarını kuvvetlendirmek ve uygulamaların geliştirme sürecini hızlandırmak için kullanılır.

* AjaxAC

AjaxAC, php tabanlı yazılmış bir kütüphanedir ve açık kaynak kodludur. Ajax uygulamaları oluşturmak ve onları geliştirmek için kullanılır. AjaxAC ile varolan PHP sınıflarına ve MySQL veritabanına, ek istek verilerini  döndürmek için geçiş oldukça kolaydır.

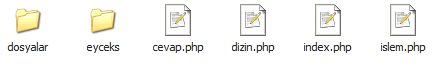
* Xajax

Xajax PHP için hazırlanmış bir kütüphanedir ve php ile birlikte Javascript, HTML, XMLHttp ve XML kullanır. Xajax ile geliştirilmiş uygulamalar,  sunucudaki php fonksiyonlarını çağırabilir ve sayfayı yeniden yüklemeden içeriğini güncelleştirebilir.  Ayrıca Xajax, promramcının Javascript hakkında ön bilgiye sahip olmasını gerektirmez.

* Zephyr  
  Zephyr, PHP5 geliştiricileri için ajax bir tabanlı kütüphanedir. Öğrenmesi ve uygulaması diğer kütüphanelere göre daha kolaydır. Smarty şablon motoru, Veri Erişim Tabakası (DAL), CPAINT uzak kütüphane ve Javascript Prototype yapısında  geliştirilmiştir.

[**AJAX İle Dosya Göndermek**](http://www.bilgisayardershanesi.com/bilgisayar_dersleri/ajax-ile-dosya-gondermek.html)

AJAX’in artık bir çok farklı alanda kullanılabileceğini eburhan‘ın AJAX Ne Gibi Durumlarda Kullanılmalı? başlıklı yazısında gördük. Gelin bunlara bir de AJAX ile sitemize dosya göndermeyi ekleyelim. Bunun nasıl yapıldığını kendi yazmış olduğum AJAX Göndermeç isimli betik ile anlatmaya çalışacağım. Konu hakkında biraz daha ipucu vermek gerekirse AJAX ile dosya göndermenin mümkün olup olmadığını göreceğiz.  
  
**AJAX ile dosya göndermenin mantığı**

Yukarıda AJAX ile dosya göndermenin mümkün olup olmadığını göreceğimizi söylemiştim. Şimdi de cevabını veriyorum; malesef hayır, yani AJAX ile dosya gönderemiyoruz. Sebebini açıklamak gerekirse, bildiğiniz gibi normal form bilgilerini GET ve POST metodlarıyla gönderiyoruz. Form ile dosya göndereceğimiz zaman da POST metodunu kullanırız ve karşı tarafta da gönderilen dosyanın bilgilerini $\_FILES[’…’] değişkeni ile alabiliriz.  
  
AJAX ile dosyayı göndermeye çalıştığımızda ise form normal bir şekilde gönderiliyor gibi görünmektedir fakat alıcı dosyada $\_FILES[’…’] değişkeni bir sonuç vermez ve hata ile karşılaşırız. Çünkü kullanmış olduğumuz tarayıcılar malesef güvenlik nedeniyle (özel ayarlar yapılmadığı müddetçe) XmlHttpRequest nesnesi üzerinden dosya göndermeye müsade etmiyor   
  
Peki bir çözümü var mıdır ve varsa nedir? Kısmen de olsa bir çözüm var diyebiliriz; gizli bir iframe kullanmak. Sitemize göndereceğimiz dosyayı seçtikten sonra formumuzu bu iframe’e göndeririz. Ardından AJAX ile dizin listemizi güncelleriz. Kullanıcı sanki tek bir işlem oluyormuş gibi görür yani dosyanın AJAX ile gönderildiğini düşünür. Oysa ki dosyayı gönderen AJAX değil bildiğimiz HTML formu ve iframe‘dir. İnanıyorum ki AJAX Göndermeç‘i açıkladıktan sonra konuyu daha iyi anlayabilirsiniz.  
  
**AJAX Göndermeç betiğimiz iki klasör ve 4 adet dosyadan oluşuyor.**  
  
**Ajax Göndermeç dosyaları**

Dosyaları tanıtmaya başlamadan önce şunu söylemeliyim ki sanırım artık dosyaların içini olduğu gibi yazmama gerek yok. Zaten makalenin sonunda indirebileceğiniz bağlantılar bulunuyor. Yazımda sadece kodlarda dikkatinizi çekmek istediğim noktaları belirteceğim.  
  
**KLASÖRLER**

dosyalar: Sitemizde göndermiş olduğumuz dosyaları barındırdığımız klasör. İsteğinize göre değiştirebilirsiniz. Önemli olan klasörün izninin yani CHMOD ayarının 777 yani yazılabilir olması gerektiği.  
  
**eyceks:**

Bu betiği hazırlarken sitemiz yazarı Erhan Burhan’ın eyceks isimli kütüphanesinin son versiyonu olan 1.3′ten faydalandım. eyceks klasöründe eyceks kütüphanesi (eyceks.js) ile yüklenme esnasında gösterilecek olan resim dosyası (loading.gif) bulunuyor.   
  
**DOSYALAR  
index.php:** Betiğimizin temel dosyası ve anasayfasıdır. Dosya gönderme formu, javascript fonksiyonlarımız ve css stil seçicilerimiz bu dosyada bulunur. Dosyanın dikkatinizi çekmek istediğim kısmı javascript fonksiyonlarıdır.  
  
PLAIN TEXTJAVASCRIPT:   
function dosyaSil(silinecekDosya) {  
if( confirm('Dosya Silinsin mi?') ) {  
var sc="yap=sil&dosya="+silinecekDosya;  
JXP(1, "dizin\_alani", "cevap.php", sc);  
}  
}  
  
function yenile() {  
var sc="yap=yenile";  
JXP(1, "dizin\_alani", "cevap.php", sc);  
}  
  
function dosyaGonder(){  
document.dosyaFormu.submit();  
var sc="yap=gonder";  
JXP(1, "dizin\_alani", "cevap.php", sc);  
document.dosyaFormu.reset();  
}   
Farkettiyseniz artık AJAX nesnesi XmlHttpRequest'i çağırmak için bir ton kod ile uğraşmıyoruz. Hazır kütüphane kullanmanın faydalarından birisi de budur. Tabi konunun mantığını anlamadan da hazır kütüphane kullanmanızı kesinlikle tavsiye etmiyorum.   
  
**Javascript fonksiyonlarına dönecek olursak:**dosyaSil(dosya) fonksiyonu cevap.php dosyasına silinmesini istediğimiz dosyanın ismini gönderir. Gelen sonuç dizin\_alani id'sine sahip alanda görüntülenir. Şunu da belirtmekte fayda var, gelen sonuç yine dizindeki dosyaları listeleme işlemidir.  
yenile() fonksiyonu dosya listesini yenilemenizi sağlayan fonksiyondur. Tüm sayfa yerine sadece AJAX sayesinde dosya listesini yenileyebiliyoruz. Yenilerken yapılan işlem cevap.php dosyasından dizini listelemesini istemekten başka bir şey değildir   
dosyaGonder() fonksiyonu önce formumuzu yani gözat diyerek seçmiş olduğunuz dosyayı iframe'e gönderir, ardından cevap.php dosyasından dosya listesini yenilemesini ister ve son olarak formu sıfırlar.

**Not:**Dosyayı iframe'e göndermesini buradan değil, formun tanım bilgisinden target="pencere" yani pencere isimli iframe diyerek belirtiyoruz.  
  
**dizin.php:** Bu dosyanın tek işlevi kendi içinde barındırdığı ve bizim düzenlediğimiz klasor değişkeni ile belirtilen dizindeki dosyaları listelemek. Bu dosyayı include('dizin.php'); diyerek anasayfada(index.php) ve cevap.php dosyasında çağırıyoruz.  
  
PLAIN TEXTPHP:   
<?php  
$klasor = "dosyalar/"; //Burayi duzenleyin  
  
$sonuc='<table style="font-size:12px;">';  
if ($dizin = opendir($klasor)) {  
while ($dosya = readdir($dizin)) {  
if ($dosya!='.' AND $dosya!='..')  
{  
$sonuc.= '<tr style="background-color:#EEEEEE;height:10px;border:1px dashed Gray;">';  
$sonuc.= '<td style="padding: 5px;" width="450" align="left">';  
$sonuc.= '<a href="dosyalar/'. $dosya. '">'. $dosya. '</a></td>'; //burasi onemli  
$sonuc.= '<td style="padding: 5px;" width="50" align="center">';  
$sonuc.= '[<a href="javascript:dosyaSil(\''. $dosya. '\');">sil</a>]</td>';  
$sonuc.= '</tr>';  
}  
}  
closedir($dizin);  
}  
$sonuc.='</table>';  
echo $sonuc;  
?>   
  
**cevap.php:**Bu dosya AJAX ile bilgileri gönderdiğimiz dosyadır. İki adet işlevi bulunmakta:

1) Eğer silme işlemi seçilmişse, dosya değişkeni ile gelen dosyayı siliyor.  
2) Dizindeki dosyaları dizin.php dosyasını çağırarak listeliyor.  
  
Burada dikkatinizi çekmek istediğim konu sleep(2) fonksiyonu. Bu fonksiyon içine girilmiş olan değer boyunca(saniye) betiği bekletir. Bekletmek istedim çünkü aynı zamanda bir yandan iframe'e gönderilmiş bir dosya bulunmakta ve bu dosya siteye yüklenmekte. Diyelim ki dosyamız biraz büyük. O halde cevap.php fonksiyonundan cevap geldiğinde dosya hala yüklenmediği için yine eski dizini görebiliriz. Bu yüzden de dosyanın gönderilmediğini sanarız. Oysa ki dosya hala gönderilmektedir. Aslında buna bir tedbir olarak anasayfaya (index.php) dizin listesini yenilemek için Yenile bağlantısı koydum ama yine de bu sleep fonksiyonu da bir önlem olabilir.   
  
PLAIN TEXTPHP:   
<?php  
  
$klasor = "dosyalar/"; //Burayi duzenleyin  
  
if ($\_POST['yap'] =="sil")   
{  
$dosyasil = $klasor.$\_POST['dosya'];  
$silindi=unlink($dosyasil);  
}  
  
sleep(2);  
  
include('dizin.php');  
?>   
  
**islem.php:**Bu dosya iframe içinde çağırdığımız dosyadır ve form ile gönderilen dosyayı yine islem.php dosyasında bulunan klasor değişkeni ile belirttiğimiz klasöre gönderir.  
  
PLAIN TEXTPHP:   
<?php  
  
$klasor = "dosyalar/"; //Burayi duzenleyin  
$giden = $klasor . basename($\_FILES['dosya']['name']);  
move\_uploaded\_file($\_FILES['dosya']['tmp\_name'], $giden)  
  
?>   
  
**Ve mutlu son**  
Bu uygulama geliştirildiği takdirde basit işlemler için güzel bir dosya yöneticisi haline dönüşebilir. Hatta dosya gönderme işlemi php'nin ftp fonksiyonlarıyla yapılırsa tıpkı bir ftp programı gibi kullanılabilen bir betik bile olabilir   
  
Umarım AJAX Göndermeç'ten memnun kalırsınız. Maksadım mantığını anlatmak olduğu için basit bir örnek sundum size. Belki ilerki seviyelerde sadece kullanmanız için güzel örnekler verebiliriz.

### Mootools ve ajax

Sayfa başında mootools.js dosyasını include edin. Basitçe ajax isteği için kod bloğu şöyle :

1. new Ajax('icerik\_alinacak\_dosya.php', {
2. method: 'get',
3. update: $('icerik\_basilacak\_element\_id')
4. }).request();

Çoğu mootools eklentisi, nesnesindeki gibi zorunlu kısımlar nesneye ilk parametrelerde veriliyor. Sonra da bir ayar dizesi (nesne olarak)  
şeklinde 2. parametre belirleniyor.  
  
Burada ilk parametre string olarak istek yapılacak url, yani istek yapılacak dosya adı. Bir php dosyası veya html dosyası veya dışarıda bir web sayfası urlsi olabilir. Ayar dizesindeki parametrelerden önemli olanların açıklamaları da şöyle :

* data Gönderilecek url parametreleri. Aşağıda açıklamasını yapacağım.
* method İstek methodu, get veya post olabilir.
* update Cevap geldiği zaman içeriğin yerleştirileceği html nesnesi adresi
* onComplete Cevap geldiği zaman işletilecek javascript kodu, fonksiyon olarak
* belirtilmesi gerekiyor ve fonksiyonun ilk parametresi cevap dönen
* içeriği string olarak fonksiyona sokar.
* evalScripts XMLHttpRequest ve çoğu diğer ajax kütüphanelerindeki ana sorun,
* cevap gelen içerik içerisinde eğer bir javascript kod bloğu veya
* dosya include varsa Normalde bu kısımlar çalıştırılmaz.
* Bu popüler bir sorundur, bu parametreye true değeri vererek
* bu sorunu aşabilirsiniz (Mootools'u seviyorum).

Şimdi bu parametreleri örneklerle açıklama çalışacağım. Önce bir bağlantıyı (link) ajax ile çağırıp dönen içeriği ekranda bir bölüme yazdıralım.

1. ...
2. **<a** href="#" onClick="ajax\_istek('hakkinda.html');"**>**Hakkında**</a>**
3. **<a** href="#" onClick="ajax\_istek('iletisim.html');"**>**iletişim**</a>**
4. ...
5. **<div** id="icerik"**></div>**
6. ...

Bu sayfamız olsun. Linke tıklantığında ajax\_istek fonksiyonunu url parametresiyle çağırıyor görüldüğü gibi. Ayrıca icerik kimlikli bir div var. İçeriği buraya basacağız.

1. function ajax\_istek(hedef){
2. new Ajax(hedef, {
3. update: $('icerik')
4. }).request();
5. }

Basit bir ajax isteği gördüğünüz gibi. Şimdi bunu biraz geliştirelim ve linke tıklandığında sayfa içeriği "Yükleniyor" olsun, içerik geldiğinde de zaten yükleniyor kalkacak.

1. function ajax\_istek(hedef){
2. $('icerik').setHTML('Yükleniyor');
3. new Ajax(hedef, {
4. update: $('icerik')
5. }).request();
6. }

Biraz daha geliştirip yüklenme işlemi bittiğinde birşeyler yaptıralım.

1. function ajax\_istek(hedef){
2. $('icerik').setHTML('Yükleniyor');
3. new Ajax(hedef, {
4. update: $('icerik'),
5. onComplete: function(){
6. alert("Sayfa Yüklendi");
7. }
8. }).request();
9. }

Örneğin burada alert yerine sayfanın bir köşesinde bu uyarıyı gösterip gizletebilirsiniz, Ya da içeriği basmayıp bir işlem yaptırabilirsiniz. Bunu örneklemek için işe biraz form katacağım. mfyz.com'da üye kayıt sayfasında kullanıcı adı kontrol mekanizmasını anlatacağım. Kayıt formuna girdiğinizde kullanıcı adı seçip yazdıktan sonra bilgi giriş kutusundan çıktığınızda (yani başka bir nesenye odaklandığınızda) otomatik olarak kullanıcı adının veritabanında olup olmadığını sorgulayıp cevabı geliyor ve ona göre kutu yeşil veya kırmızı oluyor farkında iseniz.

1. **<input** type="text" id="kadi" onBlur="kullanici\_adi\_kontrol\_et(this.value);"**>**

şeklinde bir inputum var ve onBlur olayına yani kutudan odak gittiğinde bir fonksiyon çağırıyorum.

1. function kullanici\_adi\_kontrol\_et(kadi){
2. // kullanici adini veritabanindan kontrol
3. // edecek php dosyasina sorgulatiyorum
4. new Ajax('kullanici\_adi\_kontrol.php', {
5. method : 'get',
6. data: 'kadi = ' + kadi,
7. onComplete: function(cevap){
8. if(cevap == 'OK'){
9. // kullanici adi giris kutusunun kenrlığını yeşil yapalım
10. $('kadi').setStyles({
11. 'border' : '1px solid green'
12. });
13. }else{
14. // kullanici adi giris kutusunun kenrlığını kırmızı yapalım
15. $('kadi').setStyles({
16. 'border' : '1px solid red'
17. });
18. // ve gelen hatayi basalim
19. alert(cevap);
20. }
21. }
22. }).request();
23. }

Bakın burada 2 parametreyi örnekledim. Birisi data parametresi diğeri de update kullanmak zorunda olmadığımız. Ayrıca onComplete'de tanımladığımız fonksiyonun ilk parametresi yani cevap değişkeni ajax ile dönen içeriği alıyor.  
  
Yukarıda yaptırdığım işlem özetle şöyle: kullanici\_adi\_kontrol.php dosyası get methodu ile gelen kadi değişkenindeki string'i veritabanında aratıyor. Eğer yoksa ekrana sadece "OK" basıyor. Eğer başka bir hata varsa hatayı basıyor. Mesela kullanıcı adında geçersiz karakterler var veya veritabanında o kullanıcı mevcut. Yani hata varsa "OK" dışında birşeyler oluyor. PHP hatası da olabilir.  
  
onComplete'da cevap değişkenini kontrol ettiriyorum. Eğer "OK" string'i ise kullanıcı adı yok yani ekrana başarı sinyali vermem gerek. Ben de etkileşimli olması için bilgi giriş kutusunun çerçevesini yeşil yapıyorum. Mootools ilecss özniteliklerini değiştiriyorum. Eğer OK değilse bir hata vardır. Çerçeveyi kırmızı yapıp cevap değişkenini yani dönen hatayı ekrana yazdırıyorum alert ile.  
  
Bu istekleri ve cevap işleme tekniklerini uygulamanıza göre geliştireceksiniz, Zaten burada o istek-cevap ilişkisinin boyutu önemli değil. Yani çok büyük istekler de yapabilirsiniz çok küçük de.

### Formları otomatik ajax ile post etmek

Mootools'da çok güzel bir özellik daha var. Formunuzu çok pratik bir şekilde ajax ile göndermek.

1. **<form** id="bilgi\_formu" action="kayit.php" method="post"**>**
2. Ad : **<input** type="text" name="ad"**><br>**
3. Soyad : **<input** type="text" name="soyad"**><br>**
4. **<input** type="submit" value="Gönder"**>**
5. **</form>**

böyle bir formunuz olduğunu varsayalım. Oldukça basit yani. Gönder butonuna basıldığında normal şekilde kayit.php dosyasina post methodu ile 2 değişken gidiyor biliyorsunuz. kayit.php'de de bu değişkenleri işliyorsunuz.

1. $('bilgi\_formu').send();

kodu sayesinde formu ajax ile gönderebilirsiniz. Gideceği url, methodu ve datası otomatik olarak bulunup gönderiliyor. Örnek kullanımını da şu kodlar özetleyebilirim :

1. **<form** id="bilgi\_formu" action="kayit.php" method="post"**>**
2. Ad : **<input** type="text" name="ad"**><br>**
3. Soyad : **<input** type="text" name="soyad"**><br>**
4. **<input** type="button" onClick="gonder()" value="Gönder"**>**
5. **</form>**
6. **<script>**
7. function gonder(){
8. $('bilgi\_formu').send({
9. onComplete: function(){
10. alert("Başarıyla gönderildi");
11. }
12. });
13. }
14. **</script>**

Gördüğünüz gibi submit yerine buton kullanarak bir fonksiyon çağırıyorum. ajax için kullandığımız send fonksiyonunun ilk parametresi ajax için kullandığımız ayar dizesi oluyor. Yani update ile gelen içeriği biryere yazdırabilir veya ajax için uyguladığımız taklaları burada da attırabilirsiniz.  
  
  
Mootools ajax dökümantasyonu :<http://docs.mootools.net/Remote/Ajax.js>  
Ajax Demoları :<http://demos.mootools.net/Ajax>  
<http://demos.mootools.net/Ajax.Form>

Not : Nesnelere olay ekleyip bu işleri yaptırmak daha kolaydır. Mesela form ajax ile post etmek için submiti butona çevirmek ilkel bir mantıktır. Forma doğrudan onSubmit olayı ile kural belirleyerek yaparız aslında bu işlemleri. Fakat dökümanda daha fazla kafa karışıklığı yaratmaması için doğrudan fonksiyon çağırtıp işlem yaptırdım.

[**Ajax İle Filtreleme**](http://www.bilgisayardershanesi.com/bilgisayar_dersleri/ajax-ile-filtreleme.html)

**Dosyalar**Bu uygulamada anasayfayı oluşturan index.php, filitrelemenin yapıldığı filitrele.php, oyuncuların özelliklerini göstermek istediğim oyuncu.php dosyası ve eyceks kütüphanesi içindeki dosyalar (loading.gif , eyceks.js) bulunmaktadır .   
  
Başlıyoruz . . .   
İlk olarak uygulamamıza filitrenin nasıl edileceği konusunda karar vermemiz lazım. Bu uygulamada aynı zamanda ajax'te Dinamik Formları da kullanmış olacağım. Biliyorsunuz ajax'in diğer bir güzel özelliği ise Dinamik Formlar'dır. Dinamik formdan kastım isteğimize o anda karşılık verecek olan formlarlardır. Dinamik formlarla ilgili AJAX Ne Gibi Durumlarda Kullanılmalı? başlıklı yazıya da bakabilirsiniz. Neyse konuyu fazla uzatmadan hemen ilk bölüme geçiyorum. 

**Filitreleme formumuz :**PLAIN TEXTPHP:   
<form id="form1" name="form1" method="post" action="javascript:void(0)">  
  
<select name="takimlar" onchange="hangi\_takim()">  
<option value="0">Takımlar</option>  
<option value="1">Galatasaray</option>  
<option value="2">Fenerbahçe</option>  
<option value="3">Beşiktaş</option>  
</select>  
</form>   
  
Şimdi bu formu eyceks kütüphanesi ile filitre.php'ye göndermek istiyorum. Hepimiz eyceks kütüphanesinin nasıl çalıştığını biliriz. Bunu da ek olarak yazayım.   
  
**Eyceks ile filitre.php ye gönderme :**  
PLAIN TEXTPHP:   
function hangi\_takim(){  
  
var takimlar = document.form1.takimlar.value  
  
var sc = 'takimlar='+ fc\_(takimlar)  
  
JXP(1, 'takim\_goster', 'filitrele.php', sc)  
  
}   
  
Evet şimdi sıra sistemden herhangi bir takımı seçtiğimizde bize o takımın oyuncularından bazılarını nasıl getirebileceğimizde. Bu iş filitre.php'de bitiyor tabii ki. Ben oyuncuları gosterdikten sonra bir kez daha o oyuncular arasından seçim yaptırarak oyuncu hakkında kısa ve öz bir bilgiyi ekranda göstermek istiyorum. Onun için filitre.php'de de biraz farklılık yapacağım.   
  
**filitre.php dosyasındayız :**

PLAIN TEXTPHP:   
<?  
$takim= $\_POST['takimlar'];  
?>  
  
<form id="form2" name="form2" method="post" action="javascript:void(0)">  
  
<?  
if($takim==1){  
?>  
  
<select name="oyuncular" onchange="hangi\_oyuncu()">  
<option value="">Oyuncu Seciniz..</option>  
<option value="1">Necati Ates</option>  
<option value="2">Hakan Sukur</option>   
<option value="3">Umit Davala</option>  
</select>  
  
<?  
}elseif($takim==2){  
?>  
  
<select name="oyuncular" onchange="hangi\_oyuncu()">  
<option value="">Oyuncu Seciniz..</option>  
<option value="1">Appiah </option>  
<option value="2">Serkan</option>   
<option value="3">M.Yozgatli</option>  
</select>  
  
<?  
}elseif($takim==3){  
?>  
  
<select name="takimlar" onchange="hangi\_oyuncu()">  
<option value="">Oyuncu Seciniz..</option>  
<option value="1">Vedran Runje</option>  
<option value="2">Jose Kleberson Pereira</option>  
<option value="3">Deivson Rogerio Da Silva</option>  
</select>  
  
<?  
}elseif($takim=="0"){echo "<font color=\"blue\">Herhangi Bir Takim Secmediniz !</font>";   
  
} else {echo "<font color=\"red\"><b>Hata ! </b></font>"; }  
?>  
  
</form>   
  
filitre.php'de ise şu şekilde bir olay izledim. 1. Takımı seçerse şu futbolcuları sırala, 2. takımı seçerse şu futbolcuları sırala tarzında. Yine üzerine bastırarak belirtmek istiyorum. Amaç sadece mantık. İşin içine veritabanı da girince inanılmaz güzel, zevkli ve estetik hale getirebilirsiniz olayı.   
  
filitre.php'de dikkat etmeniz gereken olay şu. Benim 3. bir aşama daha istemem. Yani Oyuncular hakkında kısa ve öz bir bilgi. Onun için tüm forumları ilk formumuz gibi düzenledim filitre.html içine çağırdıgımızda eyceks ile yine kodlar çalıştıracağız. Bu formların OnChange özelliğine dikkat edin ." hangi\_oyuncu() " Görüyorsunuz değıl mi? Şimdi filitre.html de hangi\_takim() altına bir de hangi\_oyuncu() fonksiyonu yazmamız lazım. O da şu şekilde olacak:   
  
**hangi\_oyuncu() fonksiyonu :**PLAIN TEXTPHP:   
function hangi\_oyuncu(){  
  
var oyuncular = document.form2.oyuncular.value  
  
var sc = 'oyuncular='+ fc\_(oyuncular)  
  
JXP(1, 'oyuncu\_goster', 'oyuncu.php', sc)  
  
}   
  
İşte bu fonksiyonu aynı javascript bloğumuz içine yazıyoruz Çağırdığımızda ise eyceks kütüphanesini kullananların bildiği gibi oyuncu.php'nin çıktısı, ID'si "oyuncu\_goster" olan bir DIV alanının içerisine yazdırılacak.  
  
**oyuncu.php dosyası :**

PLAIN TEXTPHP:   
<?  
$oyuncu= $\_POST['oyuncular'];  
?>  
  
<?  
if($oyuncu==1){  
?>  
  
Bu Seçtiğiniz Oyuncu 1 numaralı bir oyuncudur.  
  
<?  
}elseif($oyuncu==2){  
?>  
  
Siz şimdi 2 numaralı oyuncuyu seçtiniz.  
  
<?  
} else if ($oyuncu==3){  
?>  
  
3 Numaralı oyuncuyu seçtiniz !  
  
<?  
  
}elseif($oyuncu=="0"){echo "<font color=\"blue\">Herhangi Bir Takim Secmediniz !</font>";   
  
}else{echo "<font color=\"red\"><b>Hata ! </b></font>"; }  
  
?>

Ajax ile veri tabanından bilgi getirme

Ajax Dünyasını daha da yakından tanımak için şöyle bir örnek yapacaz hazır bir veritabanı ile ordan var olan kayıtları çekecem sizlerden benle birlikte bir veritabanı oluşturarak uygulamayı görebilirsiniz ancak veritabanı fieldlarına dikkat etmenizi ve kayıtları ona göre ayarlamanızı tavsiye ederim şimdi kodu yazalım:

<html><head>  
<script type=”text/javascript”>  
function MusteriBul(str)  
  
{  
if (str==””)  
{document.getElementById(“txtHint”).innerHTML=””;  
return;  
}  
if (window.XMLHttpRequest)

{// Bu kod şu Browserlar için IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();  
}

else

{// Bu kod ise IE6, IE5 için geçerli

xmlhttp=new ActiveXObject(“Microsoft.XMLHTTP”);

}  
xmlhttp.onreadystatechange=function()

{  
if (xmlhttp.readyState==4 && xmlhttp.status==200) // Durumlar Kontrol Ediliyor  
{  
document.getElementById(“txtHint”).innerHTML=xmlhttp.responseText;  
}  
}

xmlhttp.open(“GET”,”getcustomer.asp?q=”+str,true); //isteğin server daki yeri açılıyor  
xmlhttp.send(); //istek gönderildi  
}

</script>  
</head>  
<body>  
<form action=””>

<select name=”customers” onchange=”MusteriBul(this.value)”>

<option value=””>Bir kayit sec ve Veritabanının cevap vermesini bekle:</option>  
<option value=”ALFKI”>ilk kayıt</option>

<option value=”NORTS “>ikinci kayıt</option>

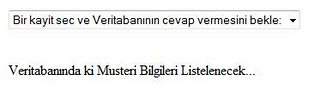
<option value=”WOLZA”>ucuncu kayıt</option>

</select>

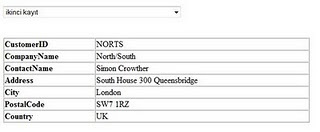
</form>  
<br />

<div id=”txtHint”>Veritabanında ki Musteri Bilgileri Listelenecek…</div>  
</body>  
</html>

Kodu derlediğimizde şu sonuç çıkacaktır:



Veritabanından yani ComboBox veya DropDownList ten bir kayıt seçiyorum ve onun bilgilerini veritabanından çağırıyorum:

[](http://salihdeveci.files.wordpress.com/2010/08/untitled38.jpg)

Yukarıda ComboBox Tool undan ikinci kayıt ı seçtiğim anda sayfa yenilenmeden ikinci kayıta ait veritabanında ki bilgiler çekildi ve tabloda yansıtıldı.